

**IN THE CLAIMS:**

1. (Original) A method, in a client, for managing connections to a server in a distributed environment, comprising:
  - establishing a connection to a server;
  - responsive to conclusion of a communication process using the connection, starting a timer;
  - maintaining a normal reference to a connection object for the connection, responsive to conclusion of a predetermined time period measured by the timer,
  - maintaining a weak reference to the connection object; and
  - periodically destroying connection objects maintained by weak references.
2. (Original) The method of claim 1, further comprising:
  - determining whether a normal reference to the connection object exists; and
  - reusing the connection if the normal reference exists.
3. (Original) The method of claim 2, further comprising restarting the timer.
4. (Original) The method of claim 1, further comprising:
  - determining whether a weak reference to the connection object exists;
  - determining whether the connection object has been destroyed if the weak reference exists; and
  - reusing the connection if the connection object has not been destroyed.
5. (Original) The method of claim 4, further comprising restarting the timer.
6. (Original) The method of claim 4, further comprising:
  - establishing a new connection if the connection object has been destroyed.
7. (Original) The method of claim 1, further comprising:

sending notification to the server that the connection object is unreferenced when a weak reference to the connection object is maintained.

8. (Original) The method of claim 1, wherein the step of periodically destroying connection objects maintained by weak references comprises destroying the connection object in response to garbage collection by the server.

9. (Original) The method of claim 1, wherein the client is a client Java Virtual Machine.

10. (Original) The method of claim 9, wherein the server is a server Java Virtual Machine.

11. (Original) The method of claim 10, wherein the client Java Virtual Machine and the server Java virtual Machine reside on the same host machine.

12. (Original) The method of claim 10, wherein the server Java virtual Machine resides on a host machine that is remote from the machine on which the client Java Virtual Machine resides.

13. (Original) A method, in a client, for reusing a connection to a server, comprising: identifying a weak reference to a connection object for a connection to a server; determining whether the connection object has been destroyed; and reusing the connection if the connection object has not been destroyed.

14. (Original) The method of claim 13, further comprising: responsive to garbage collection by the server, destroying the connection object.

15. (Original) The method of claim 13, further comprising: establishing a new connection if the connection object has been destroyed.

16. (Original) The method of claim 13, wherein the connection object is a Java object.

17. (Original) The method of claim 16, wherein the Java object is a remote method invocation object.

18. (Original) A method, in a client, for caching connections to a server, comprising:  
adding a reference to a connection object for a connection to a weak hash map  
and a hash map;

responsive to conclusion of a communication process using the connection,  
starting a timer; and

responsive to conclusion of a predetermined time period measured by the timer,  
removing the reference to the connection object from the hash map.

19. (Currently amended) The method of claim 18, wherein the connection is between  
the client and the server, and the connection object is used to enable the connection  
between the client and server, further comprising:

determining whether the connection object has been destroyed; and  
removing the reference to the connection object from the weak hash map if the  
connection object has been destroyed; and  
maintaining the reference to the connection object in the weak hash map if the  
connection object has not been destroyed to thereby allow use of such connection object  
by a subsequent communication process between the client and server without  
establishing a new connection between the client and server.

20. (Original) An apparatus, in a client, for managing connections to a server in a  
distributed environment, comprising:

connection means for establishing a connection to a server;  
timer means for starting a timer responsive to conclusion of a communication  
process using the connection;

normal reference means for maintaining a normal reference to a connection object for the connection;

weak reference means for maintaining a weak reference to the connection object responsive to conclusion of a predetermined time period measured by the timer; and

garbage collection means for periodically destroying connection objects maintained by weak references.

21. (Original) The apparatus of claim 20, further comprising:  
means for determining whether a normal reference to the connection object exists;  
and  
means for reusing the connection if the normal reference exists.
22. (Original) The apparatus of claim 21, further comprising means for restarting the timer.
23. (Original) The apparatus of claim 20, further comprising:  
means for determining whether a weak reference to the connection object exists;  
means for determining whether the connection object has been destroyed if the weak reference exists; and  
means for reusing the connection if the connection object has not been destroyed.
24. (Original) The apparatus of claim 23, further comprising means for restarting the timer.
25. (Original) The apparatus of claim 23, further comprising:  
means for establishing a new connection if the connection object has been destroyed.
26. (Original) The apparatus of claim 20, further comprising:  
means for sending notification to the server that the connection object is unreferenced when a weak reference to the connection object is maintained.

27. (Original) The apparatus of claim 20, wherein the client is a client Java Virtual Machine.

28. (Original) The apparatus of claim 27, wherein the server is a server Java Virtual Machine.

29. (Original) The apparatus of claim 28, wherein the client Java Virtual Machine and the server Java virtual Machine reside on the same host machine.

30. (Original) The apparatus of claim 28, wherein the server Java virtual Machine resides on a host machine that is remote from the machine on which the client Java Virtual Machine resides.

31. (Original) An apparatus, in a client, for reusing a connection to a server, comprising:  
identification means for identifying a weak reference to a connection object for a connection to a server;  
determination means for determining whether the connection object has been destroyed; and  
connection means for reusing the connection if the connection object has not been destroyed.

32. (Original) The apparatus of claim 31, further comprising:  
garbage collection means for destroying the connection object responsive to garbage collection by the server.

33. (Original) The apparatus of claim 31, further comprising:  
means for establishing a new connection if the connection object has been destroyed.

34. (Original) The apparatus of claim 31, wherein the connection object is a Java object.

35. (Original) The apparatus of claim 34, wherein the Java object is a remote method invocation object.

36. (Original) An apparatus, in a client, for caching connections to a server, comprising:

reference means for adding a reference to a connection object for a connection to a weak hash map and to a hash map;

timer means for starting a timer responsive to conclusion of a communication process using the connection; and

removal means for removing the reference to the connection object from the hash map responsive to conclusion of a predetermined time period measured by the timer.

37. (Currently amended) The apparatus of claim 36, wherein the connection is between the client and the server, and the connection object is used to enable the connection between the client and server, further comprising:

means for determining whether the connection object has been destroyed; and

means for removing the reference to the connection object from the weak hash map if the connection object has been destroyed; and

means for maintaining the reference to the connection object in the weak hash map if the connection object has not been destroyed to thereby allow use of such connection object by a subsequent communication process between the client and server without establishing a new connection between the client and server.

38. (Original) A computer program product, in a computer readable medium, for managing connections in a distributed environment, comprising:

instructions for establishing a connection;

instructions for starting a timer responsive to conclusion of a communication process using the connection;

instructions for maintaining a normal reference to a connection object for the connection;

instructions for maintaining a weak reference to the connection object responsive to conclusion of a predetermined time period measured by the timer; and

instructions for periodically destroying connection objects maintained by weak references.

39. (Original) A computer program product, in a computer readable medium, for reusing a connection to a server, comprising:

instructions for identifying a weak reference to a connection object for a connection;

instructions for determining whether the connection object has been destroyed; and

instructions for reusing the connection if the connection object has not been destroyed.

40. (Original) A computer program product, in a computer readable medium, for caching connections to a server, comprising:

instructions for adding a reference to a connection object for a connection to a weak hash map and a hash map;

instructions for starting a timer responsive to conclusion of a communication process using the connection; and

instructions for removing the reference to the connection object from the hash map responsive to conclusion of a predetermined time period measured by the timer.